Guide contents

Know the folder roles

Load one script

Change content carefully

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Noita modding

Noita mods are local folders with metadata, Lua hooks and optional game-data overrides. The community-maintained Noita modding basics distinguishes a mod's own `files/` from a `data/` tree used to replace existing game files. Begin with a hook that logs a message; that separates loading problems from content or gameplay problems.

> **Quick start:** create a uniquely named folder under Noita's `mods/`, add `mod.xml` and `init.lua`, enable the mod in the in-game mod menu, restart, and verify one log message before touching game data.

Know the folder roles

The modding basics show this pattern:

```text
Noita/
  mods/
    moddocs_test/
      mod.xml
      init.lua
      files/
      data/
```

`files/` holds assets unique to your mod. `data/` mirrors paths from Noita's game data when you intentionally override a vanilla file. Start without `data/` so the first experiment cannot accidentally replace unrelated content. The folder name is your local mod identifier and should avoid colliding with another installed mod. See the mod.xml reference for current metadata fields and dependency declarations.

Load one script

Noita invokes documented callbacks in `init.lua`, including `OnModPreInit`, `OnModInit`, `OnModPostInit` and `OnPlayerSpawned`. Choose `OnModInit` for a simple load check and use the Lua API reference for the installed game version when extending it. A first `init.lua` can be:

```lua
function OnModInit()
    print("ModDocs test mod initialized")
end
```

Create metadata using the current mod.xml example, set the script path to this `init.lua`, and avoid using an ID copied from another mod. Enable it through the game's Mods menu and restart as described by the installation guide. Confirm the message in the game's log. If absent, check XML syntax, the script path, whether the mod is enabled, and any errors before editing the script further.

Change content carefully

To alter an existing file, first identify its actual path and dependencies in the installed game data. Put the override at the matching path under your mod's `data/`. To extend another Lua file without replacing it wholesale, the basics document `ModLuaFileAppend`, called during initialization. Use the appropriate hook and test load order with other mods. An override may conflict with any other mod that changes the same file; append or patch only where the game's API makes that reliable.

When adding an entity, spell, perk or biome behavior, trace an existing game example and confirm every referenced XML, Lua, sprite and audio path. Test from a fresh run as well as an existing save where relevant. Keep a changelog of every vanilla file you

replace. Package only your mod folder and work you have permission to distribute; do not ship copied game archives.

This guide is **unfinished** beyond loading and structure. A complete, version-tested new spell or entity walkthrough and compatibility matrix still need to be added. Consult the linked modding wiki for the specific APIs before publishing gameplay content.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- Noita modding basics
- mod.xml reference
- Lua API reference
- installation guide